



Server-Server Protocol
XML Syntax Document
V1.0

WV Internal Tracking Number: WV-014

Notice

Copyright © 2001-2002 **Ericsson, Motorola and Nokia**. All Rights Reserved.

Implementation of all or part of any Specification may require licenses under third party intellectual property rights, including without limitation, patent rights (such a third party may or may not be a Supporter). The Sponsors of the Specification are not responsible and shall not be held responsible in any manner for identifying or failing to identify any or all such third party intellectual property rights.

THIS DOCUMENT AND THE INFORMATION CONTAINED HEREIN ARE PROVIDED ON AN "AS IS" BASIS WITHOUT WARRANTY OF ANY KIND AND ERICSSON, MOTOROLA and NOKIA DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT SHALL ERICSSON, MOTOROLA or NOKIA BE LIABLE TO ANY PARTY FOR ANY LOSS OF PROFITS, LOSS OF BUSINESS, LOSS OF USE OF DATA, INTERRUPTION OF BUSINESS, OR FOR DIRECT, INDIRECT, SPECIAL OR EXEMPLARY, INCIDENTAL, PUNITIVE OR NONSEQUENTIAL DAMAGES OF ANY KIND IN CONNECTION WITH THIS DOCUMENT OR THE INFORMATION CONTAINED HEREIN, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH LOSS OR DAMAGE. The above notice and this paragraph must be included on all copies of this document that are made.

1. INTRODUCTION	4
2. REFERENCES	4
3. DOCUMENT TYPE DEFINITION	5
4. DATA TYPE ASSIGNMENTS FOR SSP XML ELEMENTS	17
5. DATA TYPES FOR XML ELEMENTS AND ATTRIBUTES	20
REDEFINED AND NEW DATA TYPES.....	20
5.1.1 <i>Boolean</i>	20
5.1.2 <i>MSISDN</i>	20
5.1.3 <i>URL</i>	20
DATA TYPES FOR XML ATTRIBUTES.....	20
DATA TYPES FOR XML TERMINAL ELEMENTS.....	27
DESCRIPTION OF COMPOSED XML ELEMENTS.....	31
6. XML BINDING EXAMPLES	34
LOGIN TRANSACTION	34
STATUS PRIMITIVE WITH DETAILS.....	34
LOGOUT TRANSACTION	34
KEEPALIVE TRANSACTION	35
GET A VAILABLESERVICE TRANSACTION	35
SERVICEINDICATION TRANSACTION	36
SET SERVICE AGREEMENT TRANSACTION	36
GET USERPROFILE TRANSACTION.....	37
UPDATEUSERPROFILE TRANSACTION.....	37

1. INTRODUCTION

This documents defines the valid XML documents representing a valid Wireless Village SSP protocol message using Document Type Definition mechanism. The base syntactical checking can be done by a validating XML parser. Further syntactical constraints are given in the data types assignment section.

2. REFERENCES

- [\[RFC822\]](#) "Standard for the Format of ARPA Internet Text Messages", August 1982.
- [\[RFC2119\]](#) "Keywords for using RFCs to Indicate Requirements levels", Bradner, S.
- [\[RFC2778\]](#) "A Model for Presence and Instant Messaging", February 2000
[\[XML\]](#) "Extensible Markup Language 1.0 (Second Edition)", W3C recommendation, 6-October-2000
- [\[RFC2046\]](#) Borenstein N., and N. Freed, "MIME (Multipurpose Internet Mail Extensions) Part Two: Media Types", November 1996.
- [\[RFC2045\]](#) "Multipurpose Internet Mail Extensions (MIME) Part one: Format of Internet Message Bodies". Section 6.8 "Base64 Content-Transfer-Encoding".
- [\[RFC2396\]](#) Uniform Resource Identifiers (URI): Generic Syntax
- [\[PresenceAttributes\]](#) Wireless Village Presence Attributes specification.
- [\[F&F\]](#) Wireless Village Features and Functions specification.
- [\[WV-CSP\]](#) Wireless Village Client-Server Protocol, Session And Transactions specification
- [\[WV-CSP- DataTypes\]](#) Wireless Village CSP data type document

3. DOCUMENT TYPE DEFINITION

```

<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT WV-SSP-Message (SetupTransaction | Session)>
<!ATTLIST WV-SSP-Message xmlns CDATA #REQUIRED>

<!-- SetupTransaction is a transaction for session setup outside of an existing
session -->
<!ELEMENT SetupTransaction (SendSecretToken | LoginRequest | LoginResponse)>
<!ATTLIST SetupTransaction
    mode (Request | Response) #REQUIRED
    transactionID CDATA #REQUIRED>

<!-- SendSecretToken is a special request-like message without session and
MetaInfo -->
<!ELEMENT SendSecretToken (SecretToken)>
<!ATTLIST SendSecretToken
    serviceID CDATA #REQUIRED
    protocol CDATA #FIXED "WV-SSP"
    protocolVersion CDATA #FIXED "1.0">
<!ELEMENT SecretToken (#PCDATA)>
<!ATTLIST SecretToken
    encoding CDATA "base64">

<!-- LoginRequest is a special response-like message without Session and MetaInfo
-->
<!ELEMENT LoginRequest (PasswordDigest)>
<!ATTLIST LoginRequest
    serviceID CDATA #REQUIRED
    timeToLive CDATA #IMPLIED>
<!ELEMENT PasswordDigest (#PCDATA)>
<!ATTLIST PasswordDigest
    encoding CDATA "base64">

<!-- LoginResponse -->
<!ELEMENT LoginResponse (Status)>
<!ATTLIST LoginResponse
    sessionID CDATA #IMPLIED
    timeToLive CDATA #IMPLIED>

<!-- Status is contained in every normal response and it may also be a separate
message or can be sent in error cases of login phase messages -->
<!ELEMENT Status (StatusDescription?)>
<!ATTLIST Status
    code CDATA #REQUIRED>
<!ELEMENT StatusDescription (#PCDATA)>

<!-- Session -->
<!ELEMENT Session (Transaction+)>
<!ATTLIST Session
    sessionID CDATA #REQUIRED>

<!-- Transaction. It is a normal transaction inside an existing session -->
<!ELEMENT Transaction (Status | LogoutRequest | Disconnect | KeepAliveRequest |
KeepAliveResponse | GetServiceRequest | ServiceList | ServiceNegotiation |
ServiceAgreement | GetUserProfileRequest | UserProfile | UpdateUserProfileRequest
| SearchRequest | SearchResponse | StopSearchRequest | InviteRequest |
InviteResponse | CancelInviteRequest | InviteUserRequest | InviteUserResponse |
CancelInviteUserRequest | CreateContactListRequest | DeleteContactListRequest |
GetContactListRequest | GetContactListResponse | GetListMemberRequest |

```

```

AddListMemberRequest | RemoveListMemberRequest | ContactListMemberResponse |
GetListPropsRequest | SetListPropsRequest | ContactListPropsResponse |
CreateAttrListRequest | DeleteAttrListRequest | GetAttrListRequest |
GetAttrListResponse | AuthorizationRequest | AuthorizationResponse |
CancelAuthRequest | SubscribeRequest | UnsubscribeRequest | GetWatcherListRequest
| GetWatcherListResponse | PresenceNotification | GetPresenceRequest |
GetPresenceResponse | UpdatePresenceRequest | SendMessageRequest |
SendMessageResponse | ForwardMessageRequest | NewMessage | MessageDelivered |
MessageNotification | GetMessageRequest | SetMessageDeliveryMethod |
GetMessageListRequest | GetMessageListResponse | RejectMessageRequest |
DeliveryStatusRequest | DeliveryStatusReport | BlockUserRequest |
GetBlockedRequest | GetBlockedResponse | JoinGroupRequest | JoinGroupResponse |
LeaveGroupRequest | LeaveGroupIndication | GetJoinedMemberRequest |
GetJoinedMemberResponse | GetGroupMemberRequest | GetGroupMemberResponse |
AddGroupMemberRequest | RemoveGroupMemberRequest | MemberAccessRequest |
GetGroupPropsRequest | GetGroupPropsResponse | SetGroupPropsRequest |
RejectListRequest | RejectListResponse | SubscribeGroupChangeRequest |
UnsubscribeGroupChangeRequest | GetGroupSubStatusRequest |
GetGroupSubStatusResponse | GroupChangeNotice) >
<!ATTLIST Transaction
    mode                (Request | Response) #REQUIRED
    transactionID       CDATA                #REQUIRED>

<!-- MetaInfo is contained in every normal request -->
<!ELEMENT MetaInfo ( Requestor ) >
<!ATTLIST MetaInfo
    clientOriginated    ( Yes | No )         "Yes" >
<!ELEMENT Requestor ( User? ) >
<!ATTLIST Requestor
    serviceID           CDATA                #REQUIRED>
<!ELEMENT User (ClientID?)>
<!ATTLIST User
    userID              CDATA                #REQUIRED>
<!ELEMENT ClientID EMPTY >
<!ATTLIST ClientID
    url                 CDATA                #IMPLIED
    MSISDN              CDATA                #IMPLIED>

<!-- Normal PROTOCOL MESSAGES detailed. Element types used first time in the
protocol message is defined below the protocol message element type -->

<!-- LogoutRequest -->
<!ELEMENT LogoutRequest EMPTY>

<!-- Disconnect -->
<!ELEMENT Disconnect (Status?)>

<!-- KeepAliveRequest -->
<!ELEMENT KeepAliveRequest EMPTY >
<!ATTLIST KeepAliveRequest
    timeToLive          CDATA                #IMPLIED>

<!-- KeepAliveResponse -->
<!ELEMENT KeepAliveResponse (Status)>
<!ATTLIST KeepAliveResponse
    timeToLive          CDATA                #IMPLIED>

<!-- GetServiceRequest -->
<!ELEMENT GetServiceRequest EMPTY>

```

```

<!-- ServiceList -->
<!ELEMENT ServiceList (Status?, ServiceTree)>

<!-- ServiceTree -->
<!ELEMENT ServiceTree (SRV_SAP?, SRV_Common?, SRV_Presence?, SRV_IM?, SRV_Group?)

<!ELEMENT SRV_SAP (SRV_ServiceNegotiation?, SRV_UserProfileMgmt?,
SRV_ServiceRelay?)
<!ELEMENT SRV_ServiceNegotiation EMPTY>
<!ELEMENT SRV_UserProfileMgmt EMPTY>
<!ELEMENT SRV_ServiceRelay EMPTY>

<!ELEMENT SRV_Common (SRV_Invite?, SRV_ComplementaryInvite?, SRV_Search?,
SRV_VerifyUser?)>
<!ELEMENT SRV_Invite (SRV_Presence?, SRV_SharedContent?, SRV_IM?, SRV_Group?)>
<!ELEMENT SRV_Presence EMPTY>
<!ELEMENT SRV_SharedContent EMPTY>
<!ELEMENT SRV_IM EMPTY>
<!ELEMENT SRV_Group EMPTY>
<!ELEMENT SRV_ComplementaryInvite EMPTY>
<!ELEMENT SRV_Search (SRV_Group?, SRV_Other?)>
<!ELEMENT SRV_Other EMPTY>
<!ELEMENT SRV_VerifyUser EMPTY>

<!ELEMENT SRV_Presence (SRV_ContactListGet?, SRV_ContactListUpdate?,
SRV_Authorization?, SRV_WatcherList?, SRV_AttributeList?,
SRV_ContactListAddress?)>
<!ELEMENT SRV_ContactListGet EMPTY>
<!ELEMENT SRV_ContactListUpdate EMPTY>
<!ELEMENT SRV_Authorization EMPTY>
<!ELEMENT SRV_WatcherList EMPTY>
<!ELEMENT SRV_AttributeList EMPTY>
<!ELEMENT SRV_ContactListAddress EMPTY>

<!ELEMENT SRV_IM (SRV_SendMessage?, SRV_PushMessage?, SRV_MessageNotification?,
SRV_GetMessage?, SRV_SetMessageDeliveryMethod?, SRV_GetMessageList?,
SRV_RejectMessage?, SRV_DeliveryReport?, SRV_Blocking?, SRV_GroupHistory?)>
<!ELEMENT SendMessage (SRV_Group?, SRV_Contacts?)>
<!ELEMENT SRV_Contacts EMPTY>
<!ELEMENT SRV_PushMessage EMPTY>
<!ELEMENT SRV_MessageNotification EMPTY>
<!ELEMENT SRV_GetMessage EMPTY>
<!ELEMENT SRV_SetMessageDeliveryMethod EMPTY>
<!ELEMENT SRV_GetMessageList EMPTY>
<!ELEMENT SRV_RejectMessage EMPTY>
<!ELEMENT SRV_DeliveryReport EMPTY>
<!ELEMENT SRV_Blocking EMPTY>
<!ELEMENT SRV_GroupHistory EMPTY>

<!ELEMENT SRV_Group (SRV_GroupMGMT?, SRV_GetMember?, SRV_MemberMGMT?,
SRV_RejectList?)>
<!ELEMENT SRV_GroupMGMT EMPTY>
<!ELEMENT SRV_GetMember EMPTY>
<!ELEMENT SRV_MemberMGMT EMPTY>
<!ELEMENT SRV_RejectList EMPTY>

<!-- ServiceNegotiation -->
<!ELEMENT ServiceNegotiation (ServiceTree) >
<!ATTLIST ServiceNegotiation
    subProtocol          CDATA          #IMPLIED

```

```

        timeToLive          CDATA          #IMPLIED>

<!-- ServiceAgreement -->
<!ELEMENT ServiceAgreement (Status, ServiceTree )>
<!ATTLIST ServiceAgreement
        subProtocol          CDATA          #IMPLIED
        timeToLive           CDATA          #IMPLIED>

<!-- ***** -->
<!--          User Profile          -->
<!-- ***** -->

<!-- GetUserProfileRequest -->
<!ELEMENT GetUserProfileRequest (MetaInfo, UserID+)>
<!ELEMENT UserID EMPTY>
<!ATTLIST UserID
        userID          CDATA          #REQUIRED>

<!-- UserProfile -->
<!ELEMENT UserProfile (Status, UserProfileValue+)>
<!ELEMENT UserProfileValue (UPInfo+)>
<!ATTLIST UserProfileValue
        userID          CDATA          #REQUIRED>
<!ELEMENT UPInfo (#PCDATA)>
<!ATTLIST UPInfo
        attr            CDATA          #REQUIRED>

<!-- UpdateUserProfileRequest -->
<!ELEMENT UpdateUserProfileRequest (MetaInfo, UserProfileValue+)>

<!-- ***** -->
<!--          GENERAL          -->
<!-- ***** -->

<!-- SearchRequest -->
<!ELEMENT SearchRequest (MetaInfo, SearchTerm+)>
<!ATTLIST SearchRequest
        searchType      ( G | U )          #IMPLIED
        onlineStatus    ( Yes | No )      #IMPLIED
        searchLimit      CDATA             #IMPLIED
        searchID         CDATA             #IMPLIED
        searchIndex      CDATA             #IMPLIED>
<!ELEMENT SearchTerm EMPTY >
<!ATTLIST SearchTerm
        attr            CDATA             #REQUIRED
        value           CDATA             #REQUIRED>

<!-- SearchResponse -->
<!ELEMENT SearchResponse (Status, SearchResult?)>
<!ATTLIST SearchResponse
        searchID         CDATA             #IMPLIED
        searchFindings   CDATA             #REQUIRED
        searchIndex      CDATA             #REQUIRED>
<!ELEMENT SearchResult ( (User | ScreenName)* | GroupID+ )>
<!ELEMENT GroupID EMPTY >
<!ATTLIST GroupID
        groupID         CDATA             #REQUIRED>
<!ELEMENT ScreenName (#PCDATA) >
<!ATTLIST ScreenName
        groupID         CDATA             #REQUIRED>

```



```
<!-- StopSearchRequest -->
<!ELEMENT StopSearchRequest (MetaInfo)>
<!ATTLIST StopSearchRequest
    searchID          CDATA    #REQUIRED>

<!-- InviteRequest -->
<!ELEMENT InviteRequest (MetaInfo, Inviting, Invited, GroupID?, AttributeList?,
ContentIDList?, InviteNote?)>
<!ATTLIST InviteRequest
    inviteID          CDATA          #REQUIRED
    inviteType        (GR | IM | PR | SC) #REQUIRED
    validity          CDATA          #IMPLIED>
<!ELEMENT Inviting (User | ScreenName)>
<!ELEMENT Invited (User | ScreenName | ContactListID)>
<!ELEMENT ContactListID EMPTY>
<!ATTLIST ContactListID
    contactListID    CDATA          #REQUIRED>
<!ELEMENT AttributeList (Attribute*)>
<!ELEMENT Attribute EMPTY >
<!ATTLIST Attribute
    attr            CDATA          #REQUIRED>
<!ELEMENT ContentIDList (ContentID+)>
<!ELEMENT ContentID EMPTY >
<!ATTLIST ContentID
    url            CDATA          #REQUIRED>
<!ELEMENT InviteNote (#PCDATA)>

<!-- InviteResponse -->
<!ELEMENT InviteResponse (Status, Inviting, Responding, ResponseNote?)>
<!ATTLIST InviteResponse
    inviteID          CDATA          #REQUIRED
    acceptance        ( Yes | No )   #REQUIRED>
<!ELEMENT Responding (User | ScreenName | GroupID)>
<!ELEMENT ResponseNote (#PCDATA)>

<!-- InviteUserRequest -->
<!ELEMENT InviteUserRequest (MetaInfo, Inviting, Invited, GroupID?,
AttributeList?, ContentIDList?, InviteNote?)>
<!ATTLIST InviteUserRequest
    inviteID          CDATA          #REQUIRED
    inviteType        (GR | IM | PR | SC) #REQUIRED
    validity          CDATA          #IMPLIED>

<!-- InviteUserResponse -->
<!ELEMENT InviteUserResponse (Status, Inviting, Responding, ResponseNote?)>
<!ATTLIST InviteUserResponse
    inviteID          CDATA          #REQUIRED
    acceptance        ( Yes | No )   #REQUIRED>

<!-- CancelInviteRequest -->
<!ELEMENT CancelInviteRequest (MetaInfo, Canceling, Canceled, ContentIDList?,
CancelNote?)>
<!ATTLIST CancelInviteRequest
    inviteID          CDATA          #REQUIRED>
<!ELEMENT Canceling (User | ScreenName)>
<!ELEMENT Canceled (User | GroupID | ContactListID)>
<!ELEMENT CancelNote (#PCDATA)>

<!-- CancelInviteUserRequest -->
```

```
<!ELEMENT CancelInviteUserRequest (MetaInfo, Canceling, Canceled, ContentIDList?,
CancelNote?)>
<!ATTLIST CancelInviteUserRequest
    inviteID          CDATA          #REQUIRED>

<!-- VerifyUserIDRequest -->
<!ELEMENT VerifyUserIDRequest (MetaInfo, VerifyUserID+)>
<!ELEMENT VerifyUserID (DateTime?)>
<!ATTLIST VerifyUserID
    userID          CDATA          #REQUIRED>
<!ELEMENT DateTime (#PCDATA)>
<!ATTLIST DateTime
    format          CDATA          "iso8601">

<!-- VerifyUserIDResponse -->
<!ELEMENT VerifyUserIDResponse (Status, VerifyUserID*)>

<!-- ***** -->
<!-- CONTACT LIST -->
<!-- ***** -->

<!-- CreateContactListRequest -->
<!ELEMENT CreateContactListRequest (MetaInfo, ContactUser+)>
<!ATTLIST CreateContactListRequest
    contactListID   CDATA          #REQUIRED>
<!ELEMENT ContactUser EMPTY >
<!ATTLIST ContactUser
    userID          CDATA          #REQUIRED
    nick           CDATA          #REQUIRED>

<!-- DeleteContactListRequest -->
<!ELEMENT DeleteContactListRequest (MetaInfo, ContactListID+)>

<!-- GetContactListRequest -->
<!ELEMENT GetContactListRequest (MetaInfo, Version)>
<!ELEMENT Version (#PCDATA)>
<!ATTLIST Version
    format          CDATA          "DateTime">

<!-- GetContactListResponse -->
<!ELEMENT GetContactListResponse (MetaInfo, Version, ContactListID*)>
<!ATTLIST GetListMemberRequest
    defaultContactListID CDATA          #IMPLIED>

<!-- GetListMemberRequest -->
<!ELEMENT GetListMemberRequest (MetaInfo)>
<!ATTLIST GetListMemberRequest
    contactListID   CDATA          #REQUIRED>

<!-- AddListMemberRequest -->
<!ELEMENT AddListMemberRequest (MetaInfo, ContactUser+)>
<!ATTLIST AddListMemberRequest
    contactListID   CDATA          #REQUIRED>

<!-- RemoveListMemberRequest -->
<!ELEMENT RemoveListMemberRequest (MetaInfo, ContactUserSpec+)>
<!ATTLIST RemoveListMemberRequest
    contactListID   CDATA          #REQUIRED>
<!ELEMENT ContactUserSpec EMPTY >
<!ATTLIST ContactUserSpec
```

```

        userID      CDATA      #IMPLIED
        nick        CDATA      #IMPLIED>

<!-- ContactListMemberResponse -->
<!ELEMENT ContactListMemberResponse (Status, ContactUser+)>
<!ATTLIST ContactListMemberResponse
        contactListID      CDATA      #REQUIRED>

<!-- GetListPropsRequest -->
<!ELEMENT GetListPropsRequest (MetaInfo)>
<!ATTLIST GetListPropsRequest
        contactListID      CDATA      #REQUIRED>

<!-- SetListPropsRequest -->
<!ELEMENT SetListPropsRequest (MetaInfo, ContactListProperties)>
<!ATTLIST SetListPropsRequest
        contactListID      CDATA      #REQUIRED>
<!ELEMENT ContactListProperties (Property*)>
<!ELEMENT Property (#PCDATA)>
<!ATTLIST Property
        prop                CDATA      #REQUIRED>

<!-- ContactListPropsResponse -->
<!ELEMENT ContactListPropsResponse (Status, ContactListProperties)>

<!-- CreateAttrListRequest -->
<!ELEMENT CreateAttrListRequest (MetaInfo, AttributeList, ContactListID*,
UserID*)>
<!ATTLIST CreateAttrListRequest
        defaultList        ( Yes | No )      #REQUIRED>

<!-- DeleteAttrListRequest -->
<!ELEMENT DeleteAttrListRequest (MetaInfo, ContactListID*, UserID*)>
<!ATTLIST DeleteAttrListRequest
        defaultList        ( Yes | No )      #REQUIRED>

<!-- GetAttrListRequest -->
<!ELEMENT GetAttrListRequest (MetaInfo, AttributeList, ContactListID*, UserID*)>
<!ATTLIST GetAttrListRequest
        defaultList        ( Yes | No )      #REQUIRED
        exportList         ( Yes | No )      #REQUIRED>

<!-- GetAttrListResponse -->
<!ELEMENT GetAttrListResponse (Status, AttributeAssociation*,
DefaultAttributeList?)>
<!ATTLIST GetAttrListResponse
        defaultList        ( Yes | No )      #REQUIRED
        exportList         ( Yes | No )      #REQUIRED>
<!ELEMENT AttributeAssociation ( (UserID | ContactListID), AttributeList)>
<!ELEMENT DefaultAttributeList ( Attribute+ )>

<!-- ***** -->
<!-- PRESENCE -->
<!-- ***** -->

<!-- SubscribeRequest -->
<!ELEMENT SubscribeRequest (MetaInfo, UserID*, ContactListID*, AttributeList?)>

<!-- AuthorizationRequest -->
<!ELEMENT AuthorizationRequest (MetaInfo, AuthRequestTuple+)>

```

```
<!ELEMENT AuthRequestTuple (Subscribers, Authorizer, AttributeList?)>
<!ATTLIST AuthRequestTuple
    authID    CDATA    #REQUIRED>
<!ELEMENT Authorizer EMPTY>
<!ATTLIST Authorizer
    userID    CDATA    #REQUIRED>
<!ELEMENT Subscribers (UserID*)>

<!-- AuthorizationResponse -->
<!ELEMENT AuthorizationResponse (Status, AuthResponseTuple+)>
<!ELEMENT AuthResponseTuple (Authorizer, SubscriberResult+)>
<!ATTLIST AuthResponseTuple
    authID    CDATA    #REQUIRED>
<!ELEMENT SubscriberResult (UserID)>
<!ATTLIST SubscriberResult
    granted   CDATA    #REQUIRED>

<!-- CancelAuthRequest -->
<!ELEMENT CancelAuthRequest (MetaInfo, UserID*)>

<!-- UnsubscribeRequest -->
<!ELEMENT UnsubscribeRequest (MetaInfo, UserID*, ContactListID*)>
<!-- AttributeList? -->

<!-- PresenceNotification -->
<!ELEMENT PresenceNotification (MetaInfo, Subscribers, PresenceValue+, Version?)>
<!ELEMENT PresenceValue ( Presence+, Version? )>
<!ATTLIST PresenceValue
    userID    CDATA    #REQUIRED>
<!ELEMENT Presence (#PCDATA) >
<!ATTLIST Presence
    attr      CDATA    #REQUIRED>

<!-- GetWatcherListRequest -->
<!ELEMENT GetWatcherListRequest (MetaInfo)>

<!-- GetWatcherListResponse -->
<!ELEMENT GetWatcherListResponse (Status, UserID*)>

<!-- GetPresenceRequest -->
<!ELEMENT GetPresenceRequest (MetaInfo, (VerUserID | VerContactListID)+,
Attribute* )>
<!ELEMENT VerUserID (Version?)>
<!ATTLIST VerUserID
    userID    CDATA    #REQUIRED>
<!ELEMENT VerContactListID (Version?)>
<!ATTLIST VerContactListID
    userID    CDATA    #REQUIRED>

<!-- GetPresenceResponse -->
<!ELEMENT GetPresenceResponse (Status, PresenceValue*, Version?)>

<!-- UpdatePresenceRequest -->
<!ELEMENT UpdatePresenceRequest (MetaInfo, PresenceValue+ )>

<!-- ***** -->
<!--     INSTANT MESSAGE     -->
<!-- ***** -->

<!-- SendMessageRequest -->
```

```

<!ELEMENT SendMessageRequest (MetaInfo, MessageInfo, ContentData)>
<!ATTLIST SendMessageRequest
    deliveryReport      ( Yes | No )      #REQUIRED >
<!ELEMENT MessageInfo (Recipient+, Sender, DateTime)>
<!ATTLIST MessageInfo
    messageID          CDATA              #IMPLIED
    messageURI         CDATA              #IMPLIED
    contentType        CDATA              #IMPLIED
    contentSize        CDATA              #IMPLIED
    validity           CDATA              #IMPLIED>
<!ELEMENT Recipient ((UserID | ScreenName | GroupID | ContactListID),
RecipientDisplay?)>
<!ELEMENT Sender ((UserID | GroupID), SenderDisplay?)>
<!ELEMENT RecipientDisplay (UserID | ScreenName | GroupID | Name)>
<!ELEMENT Name (#PCDATA)>
<!ELEMENT SenderDisplay (UserID | ScreenName | GroupID | Name)>
<!ELEMENT ContentData (#PCDATA)>
<!ATTLIST ContentData
    contentType        CDATA              #REQUIRED
    encoding           CDATA              "base64">

<!-- SendMessageResponse -->
<!ELEMENT SendMessageResponse (Status)>
<!ATTLIST SendMessageResponse
    messageID          CDATA              #REQUIRED>

<!-- ForwardMessageRequest -->
<!ELEMENT ForwardMessageRequest (MetaInfo, Recipient+)>
<!ATTLIST ForwardMessageRequest
    messageID          CDATA              #IMPLIED
    messageURI         CDATA              #IMPLIED>

<!-- NewMessage -->
<!ELEMENT NewMessage ((MetaInfo | Status), RecipientIDs, MessageInfo,
ContentData)>
<!ATTLIST NewMessage
    messageID          CDATA              #IMPLIED
    messageURI         CDATA              #IMPLIED>
<!ELEMENT RecipientIDs (UserID+)>

<!-- MessageDelivered -->
<!ELEMENT MessageDelivered ( MetaInfo | Status )>
<!ATTLIST MessageDelivered
    messageID          CDATA              #REQUIRED>

<!-- MessageNotification -->
<!ELEMENT MessageNotification ( MetaInfo, MessageInfo, RecipientIDs )>
<!ATTLIST MessageNotification
    messageID          CDATA              #REQUIRED>

<!-- GetMessageRequest -->
<!ELEMENT GetMessageRequest ( MetaInfo )>
<!ATTLIST GetMessageRequest
    messageID          CDATA              #REQUIRED>

<!-- SetMessageDeliveryMethod -->
<!ELEMENT SetMessageDeliveryMethod ( MetaInfo )>
<!ATTLIST SetMessageDeliveryMethod
    messageID          CDATA              #REQUIRED
    deliveryMethod     (NotifyGet | Push) #REQUIRED

```

```

        acceptedContentLength  CDATA          #REQUIRED
        groupID                CDATA          #IMPLIED>

<!-- GetMessageListRequest -->
<!ELEMENT GetMessageListRequest ( MetaInfo )>
<!ATTLIST GetMessageListRequest
        groupID                CDATA          #IMPLIED
        messageCount           CDATA          #IMPLIED>

<!-- GetMessageListResponse -->
<!ELEMENT GetMessageListResponse ( Status, MessageInfo )>

<!-- RejectMessageRequest -->
<!ELEMENT RejectMessageRequest ( MetaInfo, MessageSpec* )>
<!ELEMENT MessageSpec EMPTY >
<!ATTLIST MessageSpec
        messageID              CDATA          #IMPLIED
        messageURI              CDATA          #IMPLIED>

<!-- DeliveryStatusReport -->
<!ELEMENT DeliveryStatusReport (MetaInfo, DeliveryResult, MessageInfo)>
<!ELEMENT DeliveryResult (Status)>

<!-- BlockUserRequest -->
<!ELEMENT BlockUserRequest ( MetaInfo, BlockList?, UnblockList?, GrantList?,
UngrantList? )>
<!ATTLIST BlockUserRequest
        blockListStatus        (Active | Inactive) #REQUIRED
        grantListStatus        (Active | Inactive) #REQUIRED>
<!ELEMENT BlockList (UserID+)>
<!ELEMENT UnblockList (UserID+)>
<!ELEMENT GrantList (UserID+)>
<!ELEMENT UngrantList (UserID+)>

<!-- GetBlockedRequest -->
<!ELEMENT GetBlockedRequest ( MetaInfo )>

<!-- GetBlockedResponse -->
<!ELEMENT GetBlockedResponse ( Status, BlockList, GrantList )>
<!ATTLIST GetBlockedResponse
        blockListStatus        (Active | Inactive) #REQUIRED
        grantListStatus        (Active | Inactive) #REQUIRED>

<!-- ***** -->
<!--          GROUP          -->
<!-- ***** -->

<!-- CreateGroupRequest -->
<!ELEMENT CreateGroupRequest (MetaInfo, GroupProperties)>
<!ATTLIST CreateGroupRequest
        groupID                CDATA          #REQUIRED>
<!ELEMENT GroupProperties (Property+, WelcomeNote?)>
<!ELEMENT WelcomeNote (#PCDATA)>

<!-- DeleteGroupRequest -->
<!ELEMENT DeleteGroupRequest (MetaInfo)>
<!ATTLIST DeleteGroupRequest
        groupID                CDATA          #REQUIRED>

<!-- JoinGroupRequest -->

```

```
<!ELEMENT JoinGroupRequest (MetaInfo, ScreenName?)>
<!ATTLIST JoinGroupRequest
    groupID          CDATA          #REQUIRED
    joinedListRequest ( Yes | No )  #REQUIRED>

<!-- JoinGroupResponse -->
<!ELEMENT JoinGroupResponse (Status, JoinedList, WelcomeNote?)>
<!ELEMENT JoinedList (Name*)>

<!-- LeaveGroupRequest -->
<!ELEMENT LeaveGroupRequest (MetaInfo, GroupID)>
<!ATTLIST LeaveGroupRequest
    groupID          CDATA          #REQUIRED>

<!-- LeaveGroupIndication -->
<!ELEMENT LeaveGroupIndication ((MetaInfo | Status), ReasonText)>
<!ELEMENT ReasonText (#PCDATA)>
<!ATTLIST LeaveGroupIndication
    groupID          CDATA          #REQUIRED>

<!-- GetJoinedMemberRequest -->
<!ELEMENT GetJoinedMemberRequest (MetaInfo)>
<!ATTLIST GetJoinedMemberRequest
    groupID          CDATA          #REQUIRED>

<!-- GetJoinedMemberResponse -->
<!ELEMENT GetJoinedMemberResponse (Status, JoinedUser*)>
<!ELEMENT JoinedUser (#PCDATA)>
<!ATTLIST JoinedUser
    userID          CDATA          #REQUIRED>

<!-- GetGroupMemberRequest -->
<!ELEMENT GetGroupMemberRequest (MetaInfo)>
<!ATTLIST GetGroupMemberRequest
    groupID          CDATA          #REQUIRED>

<!-- GetGroupMemberResponse -->
<!ELEMENT GetGroupMemberResponse (Status, Admins?, Moderators?, OrdinaryUsers?)>
<!ELEMENT Admins (UserID*)>
<!ELEMENT Moderators (UserID*)>
<!ELEMENT OrdinaryUsers (UserID*)>

<!-- AddGroupMemberRequest -->
<!ELEMENT AddGroupMemberRequest (MetaInfo, UserID*)>
<!ATTLIST AddGroupMemberRequest
    groupID          CDATA          #REQUIRED>

<!-- RemoveGroupMemberRequest -->
<!ELEMENT RemoveGroupMemberRequest (MetaInfo, UserID+)>
<!ATTLIST RemoveGroupMemberRequest
    groupID          CDATA          #REQUIRED>

<!-- MemberAccessRequest -->
<!ELEMENT MemberAccessRequest (MetaInfo, Admins?, Moderators?, OrdinaryUsers?)>
<!ATTLIST MemberAccessRequest
    groupID          CDATA          #REQUIRED>

<!-- GetGroupPropsRequest -->
<!ELEMENT GetGroupPropsRequest (MetaInfo)>
<!ATTLIST GetGroupPropsRequest
```

```

        groupID                CDATA                #REQUIRED>

<!-- GetGroupPropsResponse -->
<!ELEMENT GetGroupPropsResponse (Status, GroupProperties, OwnProperties)>
<!ELEMENT OwnProperties (Property+)>

<!-- SetGroupPropsRequest -->
<!ELEMENT SetGroupPropsRequest (GroupProperties?, OwnProperties?)>
<!ATTLIST SetGroupPropsRequest
        groupID                CDATA                #REQUIRED>

<!-- RejectListRequest -->
<!ELEMENT RejectListRequest (MetaInfo, AddUsers, RemoveUsers)>
<!ATTLIST RejectListRequest
        groupID                CDATA                #REQUIRED>
<!ELEMENT AddUsers (UserID*)>
<!ELEMENT RemoveUsers (UserID*)>

<!-- RejectListResponse -->
<!ELEMENT RejectListResponse (Status, RejectList)>
<!ELEMENT RejectList (UserID*)>

<!-- SubscribeGroupChangeRequest -->
<!ELEMENT SubscribeGroupChangeRequest (MetaInfo)>
<!ATTLIST SubscribeGroupChangeRequest
        groupID                CDATA                #REQUIRED>

<!-- UnsubscribeGroupChangeRequest -->
<!ELEMENT UnsubscribeGroupChangeRequest (MetaInfo)>
<!ATTLIST UnsubscribeGroupChangeRequest
        groupID                CDATA                #REQUIRED>

<!-- GetGroupSubStatusRequest -->
<!ELEMENT GetGroupSubStatusRequest (MetaInfo)>
<!ATTLIST GetGroupSubStatusRequest
        groupID                CDATA                #REQUIRED>

<!-- GetGroupSubStatusResponse -->
<!ELEMENT GetGroupSubStatusResponse (Status)>
<!ATTLIST GetGroupSubStatusResponse
        groupID                CDATA                #REQUIRED
        subscribed             (Yes | No)         #REQUIRED>

<!-- GroupChangeNotice -->
<!ELEMENT GroupChangeNotice (Subscribers, Joined?, Left?, GroupProperties?,
OwnProperties?)>
<!ATTLIST GroupChangeNotice
        groupID                CDATA                #REQUIRED>
<!ELEMENT Joined (Name*)>
<!ELEMENT Left (Name*)>

```


4. DATA TYPE ASSIGNMENTS FOR SSP XML ELEMENTS

In this section there will be given the mapping of the informational elements in SSP primitives to XML element constructions or attributes. If the mapped value is an element construction, then the specified element construction inside the transaction element represents the SSP information element. If the mapped value is an attribute, then the information element is represented by the specified attribute of the container element (the transaction element).

Information Element	XML Element	XML Attribute
Accepted-Content-Length		acceptedContentLength
Add-User-ID-List	AddUsers	
Agreed-Service-List	ServiceTree	
Agreed-Sub-Protocol		subProtocol
Agreed-Time-To-Live		timeToLive
Attribute-Association-List	AttributeAssociation+	
Authorization-Result		granted (of SubscriberResult)
Authorizing-User-ID	Authorizer	
Block-List-Status		blockListStatus
Block-User-ID-List	BlockList	
Canceled-Content-ID-List	ContentIDList	
Canceled-User-ID-List	UserID+	
Canceled-User	Canceled	
Canceling-User	Canceling	
Cancel-Reason	CancelNote	
Client-ID		clientID
Client-Originated		clientOriginated
Contact-List-ID		contactListID
Contact-List-ID-List	ContactListID+	
Contact-List-ID-List of GetPresenceRequest	VerContactListID+	
Contact-List-Props	ContactListProperties	
Contact-List-Version	Version	
Content	ContentData	
Default-Association-List	DefaultAttributeList	
Default-C-List-ID		defaultContactListID
Default-List		defaultList
Delivery-Result	DeliveryReport	
Delivery-Report-Request		deliveryReport
Desired-Service-List	ServiceTree	
Desired-Sub-Protocol		subProtocol
Export-List		exportList
Grant-List-Status		grantListStatus
Grant-User-ID-List	GrantList	
Group-ID		groupID
Group-Prop-List	GroupProperties	
Invite-Acceptance		acceptance
Invite-Content-ID-List	ContentIDList	
Invite-Group-ID	GroupID	
Invite-ID		inviteID
Invite-Presence-Attribute-List	AttributeList	

Invite-Reason	InviteNote	
Invite-Response	ResponseNote	
Invite-Type		inviteType
Invited-User	Invited	
Inviting-User	Inviting	
Joined-Request		joinedListRequest
Joined-User-Screen-Name-List	JoinedUser+	
Joined-User-Screen-Name-List of GroupChangeNotice	Joined	
Left-User-Screen-Name-List	Left	
List-of-Subscribing-User-ID-and-Presence-Attribute-List	AuthRequestTuple/Subscribers and AuthRequestTuple/AttributeList	
Message-Count		messageCount
Message-Delivery-Method		deliveryMethod
Message-ID		messageID
Message-ID-List		messageID of MessageSpec+
Message-Info	MessageInfo	
Message-URI		messageURI
Message-URI-List		messageID of MessageSpec+
Meta-Information	MetaInfo	
Online-Status		onlineStatus
Own-Prop-List	OwnProperties	
Password-Digest	PasswordDigest	
Presence-Attribute-List	AttributeList	
Presence-Value-List	PresenceValue+	
Presence-Value-Version	Version	
Presence-Value-Version-List of GetPresenceRequest	Version of (VerUserID and VerContactListID)	
Presence-Value-Version-List	Version of (PresenceValue+)	
Protocol		protocol
Protocol-Version		protocolVersion
Reason-Text	ReasonText	
Recipients	Recipient+	
Recipient-UserID-List	RecipientIDs	
Reject-User-ID-List	RejectList	
Remove-User-ID-List	RemoveUsers	
Responding-User	Responding	
Screen-Name	ScreenName	
Search-Criteria	SearchCriteria	
Search-Findings		searchFindings
Search-ID		searchID
Search-Index		searchIndex
Search-Limit		searchLimit
Search-Pair-List	SearchTerm+	
Search-Results	SearchResult	
Search-Type		searchType
SecretToken	SecretToken	

Service-ID		serviceID
Service-List	ServiceTree	
Session-ID		sessionID
Status code		code
Status description		statusDescription
Status-Info	Status	
Subscribing-User-ID-List	Subscribers	
Subscribing-User-ID-List in AuthorizationResponse	SubscriberResult+	
Subscription-Status		subscribed
Time-To-Live		timeToLive
Transaction-ID		transactionID
Unblock-User-ID-List	UnblockList	
Ungrant-User-ID-List	UngrantList	
Updated-User-Profile-List	UserProfileValue+	
User-ID		userID
User-ID-List	UserID+	
User-ID-List of GetPresenceRequest	VerUserID+	
User-ID-List of (GetGroupMemberResponse or MemberAccessRequest)	OrdinaryUsers	
User-ID-List-Adm	Admins	
User-ID-List-Mod	Moderators	
User-List	ContactUser+	
User-List in RemoveListMemberRequest	ContactUserSpec+	
User-Profile-List	UserProfileValue+	
Validity		validity
Verify-User-ID-List	VerifyUserID+	
Welcome-Text	WelcomeNote	

5. DATA TYPES FOR XML ELEMENTS AND ATTRIBUTES

Redefined and new data types

5.1.1 Boolean

A boolean value indicates either true or false.

It is encoded to XML as an enumerated type with possible values of:

- "Yes" : means the specified value is true.
- "No" : means the specified value is false.

These values are case sensitive. Other values are invalid.

5.1.2 MSISDN

It is a string containing a valid MSISDN number.

5.1.3 URL

The value of the type is a string of a Uniform Resource Locator. See rfc2396.

Data types for XML Attributes

In this section we will define the XML attributes of the used elements. Description of thre basic types can be found in [WV-DataTypes].

XML Attribute	acceptance
Data type	Boolean
Description	Specifies if the invited party accepted the invitation.

XML Attribute	acceptedContentLength
Data type	Integer
Description	Maximum size of message that can be pushed to the user. See [WV-SSP] Section 12.2.8.

XML Attribute	attr
Data type	String
Description	Defines an attribute name. It can be either a presence attribute or a user profile attribute depending on the container element.

XML Attribute	authID
Data type	String

Description	Gives an identifier for an authentication request tuple. It is specified by the requestor server. Unique for a session.
-------------	---

XML Attribute	blockListStatus
Data type	Enum
Range	"Active" "Inactive"
Description	Indicates if the block list is in use ("Active") or not ("Inactive"). See [WV-SSP] Section 12.2.14.

XML Attribute	clientOriginated
Data type	Boolean
Description	Attribute of the Transaction method. It specifies whether the transaction is originated from a client (Yes) or from a Wireless Village server (No).

XML Attribute	code
Data type	Integer
Description	Defined only in the Status element. It specifies the status code. Possible status codes are defined in Section 9. of the [WV-CSP].

XML Attribute	contactListID
Data type	String
Description	Corresponds to the Contact-List-ID defined in [WV-SSP] Section 3.3.3.

XML Attribute	contentType
Data type	String
Description	Specifies the content type of an instant message. Must be in MIME type format.

XML Attribute	defaultList
Data type	Boolean
Description	Specifies if the container attribute list operation element is for the default list of the user ("Yes") or for an ordinary list ("No").

XML Attribute	deliveryMethod
---------------	----------------

Data type	Enum
Range	"NotifyGet" "Push"
Description	Specifies the method used for message delivery. See [WV-SSP] and [WV-CSP] Section 7.1.2.1.

XML Attribute	deliveryReport
Data type	Boolean
Description	Specifies if a delivery report is requested for the sended instant message.

XML Attribute	encoding
Data type	String
Description	Defines the used encoding method for the content of the element. Encoding scheme is base64 by default (i.e. when this attribute is not specified).

XML Attribute	exportList
Data type	Boolean
Description	See [WV-SSP].

XML Attribute	format
Data type	String
Description	Defines the format of the container element. In this version this value is only specified for Version, for what the default format is DateTime. This format means a value in DateTime format specified in [WV-DataTypes].

XML Attribute	granted
Data type	Boolean
Description	Specifies for a user if his authentication request was allowed ("Yes") or not ("No").

XML Attribute	grantListStatus
Data type	Enum
Range	"Active" "Inactive"

Description	Indicates if the grant list is in use ("Active") or not ("Inactive"). See [WV-SSP] Section 12.2.14.
-------------	---

XML Attribute	groupID
Data type	String
Description	Corresponds to the Group-ID defined in [WV-SSP] Section 3.3.4.

XML Attribute	inviteID
Data type	String
Description	See in [WV-SSP].

XML Attribute	inviteType
Data type	Enum
Range	"GR" "IM" "PR" "SC"
Description	Defines the type of the invitation. In case of value "GR", the invitation is for a group, in case of value "IM", the invitation is for instant messaging, in case of "PR", the invitation is for a presence attribute, in case of "SC" the invitation is for a shared content.

XML Attribute	joinedListRequest
Data type	Boolean
Description	See in [WV-SSP] Section 13.1.3.

XML Attribute	messageCount
Data type	Integer
Description	The maximum number of message-info structures to be returned.

XML Attribute	messageID
Data type	String
Description	Defines a message identifier that uniquely identifies the message. The generator of the message must guarantee that the message identifier is globally unique. Must be in format msg-id of rfc2822. For the right-hand side of the msg-id, the domain name of the server should be used.

XML Attribute	messageURI
---------------	------------

Data type	URI
Description	Specifies a message by a Uniform Resource Locator.

XML Attribute	mode
Data type	Integer
Range	"Request" "Response"
Description	Attribute of the Transaction method. It specifies whether the transaction is a request of a response without parsing of the content of the transaction.

XML Attribute	MSISDN
Data type	MSISDN
Description	Specifies a valid phone number, by which for example a client can be identified.

XML Attribute	nick
Data type	String
Description	Defines a nickname for a user on the contact list.

XML Attribute	onlineStatus
Data type	Boolean
Description	Specifies for a search whether the search is only for online users.

XML Attribute	prop
Data type	String
Description	Defines a property name.

XML Attribute	protocol
Data type	String
Range	"WV-SSP"
Description	Defines the procotol used. Must be "WV-SSP".

XML Attribute	protocolVersion
Data type	String
Range	"1.0"
Description	Defines the version of the procotol used. Must be "1.0".

XML Attribute	searchFindings
Data type	Integer
Description	See in [WV-SSP].

XML Attribute	searchID
Data type	String
Description	See in [WV-SSP].

XML Attribute	searchIndex
Data type	Integer
Description	See in [WV-SSP].

XML Attribute	searchLimit
Data type	Integer
Description	See in [WV-SSP].

XML Attribute	searchType
Data type	Enum
Range	"G" "U"
Description	Defines the type of the search. In case of value "G", the search is for a group, in case of value "U", the search is for users.

XML Attribute	serviceID
Data type	String
Format	Defined in [WV-SSP] Section 3.3.7.
Description	See [WV-SSP] Section 3.3.7.

XML Attribute	sessionID
Data type	String
Description	Specifies the used session identifier.

XML Attribute	subscribed
Data type	Boolean

Description	Used in a subscription status response. Specifies if the asked entity is subscribed ("Yes") to the group or not ("No").
-------------	---

XML Attribute	subProtocol
Data type	String
Description	During session negotiation it specifies the subprotocol name and version for proprietary extensions.

XML Attribute	timeToLive
Data type	Integer
Description	Defines the time in second how long the containing action is valid.

XML Attribute	url
Data type	URL
Description	Identifies a content or a client by a Uniform Resource Locator.

XML Attribute	userID
Data type	String
Description	Corresponds to the User-ID defined in [WV-SSP] Section 3.3.2

XML Attribute	validity
Data type	Integer
Description	Specifies the time in seconds how long the specified entity is regarded valid.

XML Attribute	value
Data type	String
Description	Specifies the value for the attribute and the operand of the search to be executed.

XML Attribute	transactionID
Data type	String
Description	Identifies a SSP transaction. The initiating party assigns this ID.

XML Attribute	xmlns
Data type	URL
Description	Specifies the namespace the message is using.

Data types for XML Terminal Elements

In this section we will define the SSP specific or redefined XML elements. Description of the basic types and description of terminal elements common with CSP can be found in [WV-DataTypes]. If you find a XML terminal element definition in both this document and in [WV-DataTypes] then the given XML terminal elements is redefined in this document and you have to use the definition below. Empty terminal elements representing protocol primitives are not listed here.

For description of the XML terminal elements inside the ServiceTree element see the Section 7.1 of [WV-SSP] document.

XML ELEMENT	Attribute
Data type	EMPTY
Format	
Description	Container element for attr attribute.
Range	

Table 1. Attribute

XML ELEMENT	Authorizer
Data type	EMPTY
Format	
Description	Defines an authorizer user using userid.
Range	

Table 2. Authorizer

XML ELEMENT	CancelNote
Data type	String
Format	
Description	Short descriptive text for canceling of invitation.
Range	

Table 3. CancelNote

XML ELEMENT	ClientID
Data type	EMPTY
Format	
Description	Container element for possible client identification methods (url, msisdh etc..)
Range	

Table 4. ClientID

XML ELEMENT	ContactListID
Data type	EMPTY
Format	
Description	Container element for contactListID attribute.
Range	

Table 5. ContactListID

XML ELEMENT	ContactUser
Data type	EMPTY
Format	
Description	Defines a user in a contact list by user id and nickname.
Range	

Table 6. ContactUser

XML ELEMENT	ContactUserSpec
Data type	EMPTY
Format	
Description	Uniquely identifies a user in the contact list either by user id or nickname.
Range	

Table 7. ContactUserSpec

XML PCDATA	ContentID
Data type	EMPTY
Format	Defined in [WV-CSP].
Description	Identifies a shared content by url attribute.
Range	

Table 8. ContentID

XML PCDATA	DateTime
Data type	Date and Time
Format	Specified in the forat attribute.
Description	Date and time.
Range	

Table 9. DateTime

XML ELEMENT	GroupID
Data type	EMPTY
Format	
Description	Container element for groupID attribute.
Range	

Table 10. GroupID

XML ELEMENT	JoinedUser
Data type	String
Format	
Description	Specifies a joined user of a group with screen name and UserID. The element contains the actual screen name of the user in the specified group, the userID attribute contains the UserID.
Range	

Table 11. JoinedUser

XML ELEMENT	MessageSpec
Data type	EMPTY
Format	
Description	Uniquely identifies a message by message id or url.
Range	

Table 12. MessageSpec

XML ELEMENT	PasswordDigest
Data type	Binary
Format	Defined in encoding attribute. BASE64 by default.
Description	The password digest used in login phase. See [WV-SSP]
Range	

Table 13. PasswordDigest

XML ELEMENT	Presence
Data type	String
Format	
Description	Text containing a user presence information identified by the attr attribute and its value is contained in the element value.
Range	

Table 14. Presence

XML ELEMENT	Property
Data type	String
Format	
Description	Text containing a property information identified by the prop attribute and its value is contained in the element value.
Range	

Table 15. Property

XML ELEMENT	ReasonText
Data type	String
Format	
Description	Text describing a reason of an operation.
Range	

Table 16. ReasonText

XML ELEMENT	SearchTerm
Data type	EMPTY
Format	
Description	Container element for search terms. Contains an attribute type (attr) and a value which the search operation must be carried out.
Range	

Table 17. SearchTerm

XML ELEMENT	ScreenName
Data type	String
Format	
Description	Defines the screen name of a user in a group. Group is specified in the groupID attribute.
Range	

Table 18. ScreenName

XML ELEMENT	SecretToken
Data type	Binary
Format	Defined in encoding attribute. By default BASE64 encoded
Description	Secret token used in login phase.
Range	

Table 19. SecretToken

XML ELEMENT	StatusDescription
Data type	String
Format	
Description	Text containing a status description in response of a request.
Range	

Table 20. StatusDescription

XML ELEMENT	UPInfo
Data type	String

Format	
Description	Text containing a user profile information indetified by the attr attribute and its value is contained in the element value.
Range	

Table 21. UPInfo

XML ELEMENT	UserID
Data type	EMPTY
Format	
Description	Container element for userID attribute.
Range	

Table 22. UserID

XML ELEMENT	Version
Data type	EMPTY
Format	
Description	Specifies a version in the element vale. Format specified in the format attribute. By default format is DateTime defined in [WV-CSP].
Range	

Table 23. Version

XML ELEMENT	WelcomeNote
Data type	String
Format	
Description	Welcome text of a group.
Range	

Table 24. WelcomeNote

Description of composed XML Elements

In this section there will be given the descriptions of the composed XML elements neither assigned to an informational element nor beeing terminal elements.

XML Element	AuthRequestTuple
Description	This element contains the Authorizer (Authorizing-User-ID informational element), the Subscribers (List-of-Subscribing-User-ID informational element) and the AttributeList (Presence-Attribute-List informational element). It means who (Subscribers) want what information (AttributeList) about which user (Authorizer). One AuthorizationRequest primitive can contain more AuthRequestTuples. Different AuthRequestTuples represent different authorization requests.

XML Element	AuthResponseTuple
Description	This element contains the Authorizer (Authorizing-User-ID informational element) and the SubscriberResults (Subscribing-User-ID-List and Authorization-Result informational elements together). It means who responds to the request (Authorizer) and who he allows or not (SubscriberResult). One AuthorizationResponse primitive can contain more AuthRequestTuples. Different AuthResponseTuples represent different authorization responses.

XML Element	PresenceValue
Description	Contains a list of presence attributes along with its values (Presence-Value-List informational element) and an optional version information (Presence-Value-Version informational element)

XML Element	Recipient
Description	Defines the recipient of an instant message. The instant message must be delivered to the address defined in this field. Can contain an optional RecipientDisplay element describing what destination should be displayed to the recipient user (e.g. the group name where the message was originally sent).

XML Element	Requestor
Description	Represents the requestor of a request primitive.

XML Element	Sender
Description	Describes the sender of an instant message. Can contain the optional SenderDisplay element describing what source should be displayed to the recipient user.

XML Element	Session
Description	Session element represents transactions sended in the same sessions. Its sessionID attribute identifies the session between the requestor and the provider server.

XML Element	SetupTransaction
Description	Contains a transaction outside of n existing session. Possible child elements are the transactions for session setup.

XML Element	SubscriberResult
Description	Defines the result of one authorization request. Contains the UserID who is allowed/forbidden to access the defined attribute and the granted attribute that defines if the access was allowed or not.

XML Element	Transaction
Description	Contains an SSP transaction primitive.

XML Element	User
Description	Represents a specific user using a specific terminal.

XML Element	VerUserID
Description	Represents a versioned user identifier (User-ID-List informational element). See description of Presence-Value-Version-List in [WV-SSP].

XML Element	VerContactListID
Description	Represents a versioned contact list identifier (Contact-List-ID-List informational element). See description of Presence-Value-Version-List in [WV-SSP].

XML Element	WV-SSP-Message
Description	Represents the SSP protocol message

6. XML BINDING EXAMPLES

In this sector some examples of selected SSP transactions will be presented.

Login Transaction

```
<!-- Message from ServerA to ServerB -->
<WV-SSP-Message xmlns="http://www.wireless-village.org/SSP1.0">
  <SetupTransaction mode="Request" transactionID="0">
    <SendSecretToken serviceID="wv@operator.hu" protocol="WV-SSP"
    procotolVersion="1.0">
      <SecretToken encoding="base64">
        R5R5FHJF47RY838289290050W0R989E0ER0R9E0392848GFJF8484JF84839294U745723934U7
      </SecretToken>
    </SendSecretToken>
  </SetupTransaction>
</WV-SSP-Message>
```

```
<!-- Message from ServerB to ServerA -->
<WV-SSP-Message xmlns="http://www.wireless-village.org/SSP1.0">
  <SetupTransaction mode="Response" transactionID="0">
    <LoginRequest serviceID="wv@operator.hu" timeToLive="1314000">
      <PasswordDigest
encoding="base64">FJS823HFUGF73HFBNE73BCJED34E60F5504W560001929394
E52</PasswordDigest>
    </LoginRequest>
  </SetupTransaction>
</WV-SSP-Message>
```

```
<!-- Message from ServerA to ServerB -->
<WV-SSP-Message xmlns="http://www.wireless-village.org/SSP1.0">
  <SetupTransaction mode="Response" transactionID="0">
    <LoginResponse sessionID="47563@someone.za" timeToLive="1314000">
      <Status code="200"/>
    </LoginResponse>
  </SetupTransaction>
</WV-SSP-Message>
```

Status primitive with details

```
<WV-SSP-Message xmlns="http://www.wireless-village.org/SSP1.0">
<Session sessionID="123456@pgsm.hu">
  <Transaction mode="Response" transactionID="420042">
    <Status code="400">
      <StatusDescription>Bad request</StatusDescription>
    </Status>
  </Transaction>
  <Transaction mode="Response" transactionID="420043">
    <Status code="200"/>
  </Transaction>
</Session>
</WV-SSP-Message>
```

Logout Transaction

```
<WV-SSP-Message xmlns="http://www.wireless-village.org/SSP1.0">
```

```
<Session sessionID="123456@someone.za">
  <Transaction mode="Request" transactionID="5006">
    <LogoutRequest/>
  </Transaction>
</Session>
</WV-SSP-Message>

<WV-SSP-Message xmlns="http://www.wireless-village.org/SSP1.0">
<Session sessionID="123456@operator.hu">
  <Transaction mode="Response" transactionID="5006">
    <Disconnect>
      <Status code="200"/>
    </Disconnect>
  </Transaction>
</Session>
</WV-SSP-Message>
```

KeepAlive transaction

```
<WV-SSP-Message xmlns="http://www.wireless-village.org/SSP1.0">
<Session sessionID="123456@someone.za">
  <Transaction mode="Request" transactionID="5003">
    <KeepAliveRequest timeToLive="3600"/>
  </Transaction>
</Session>
</WV-SSP-Message>

<WV-SSP-Message xmlns="http://www.wireless-village.org/SSP1.0">
<Session sessionID="123456@operator.hu">
  <Transaction mode="Response" transactionID="5003">
    <KeepAliveResponse timeToLive="3600">
      <Status code="200"/>
    </KeepAliveResponse>
  </Transaction>
</Session>
</WV-SSP-Message>
```

GetAvailableService transaction

```
<WV-SSP-Message xmlns="http://www.wireless-village.org/SSP1.0">
<Session sessionID="123456@someone.za">
  <Transaction mode="Request" transactionID="5002">
    <GetServiceRequest/>
  </Transaction>
</Session>
</WV-SSP-Message>

<WV-SSP-Message xmlns="http://www.wireless-village.org/SSP1.0">
<Session sessionID="123456@operator.hu">
  <Transaction mode="Response" transactionID="5002">
    <ServiceList>
      <Status code="200"/>
    <ServiceTree>
      <SRV_SAP/>
    </ServiceTree>
  </Transaction>
</Session>
</WV-SSP-Message>
```

```
<SRV_Common/>
<SRV_Presence/>
<SRV_IM/>
<SRV_Group/>
<ServiceTree>
</ServiceList>
</Transaction>
</Session>
</WV-SSP-Message>
```

ServiceIndication transaction

```
<WV-SSP-Message xmlns="http://www.wireless-village.org/SSP1.0">
<Session sessionID="123456@operator.hu">
<Transaction mode="Response" transactionID="5002">
<ServiceList>
<Status code="200"/>
<ServiceTree>
<SRV_SAP/>
<SRV_Common/>
<SRV_Presence/>
<SRV_IM/>
<SRV_Group/>
<ServiceTree>
</ServiceList>
</Transaction>
</Session>
</WV-SSP-Message>
```

SetServiceAgreement transaction

```
<WV-SSP-Message xmlns="http://www.wireless-village.org/SSP1.0">
<Session sessionID="123456@someone.za">
<Transaction mode="Request" transactionID="113">
<ServiceNegotiation timeToLive="3600">
<ServiceTree>
<SRV_SAP/>
<SRV_Common/>
<SRV_Presence/>
<SRV_IM/>
<SRV_Group/>
<ServiceTree>
</ServiceNegotiation>
</Transaction>
</Session>
</WV-SSP-Message>
```

```
<WV-SSP-Message xmlns="http://www.wireless-village.org/SSP1.0">
<Session sessionID="123456@operator.hu">
<Transaction mode="Response" transactionID="113">
<ServiceAgreement timeToLive="3600">
<Status code="200"/>
<ServiceTree>
<SRV_SAP/>
<SRV_Common/>
<SRV_Presence/>
```

```
<SRV_IM/>
  <SRV_Group/>
  <ServiceTree>
</ServiceAgreement>
</Transaction>
</Session>
</WV-SSP-Message>
```

GetUserProfile transaction

```
<WV-SSP-Message xmlns="http://www.wireless-village.org/SSP1.0">
<Session sessionID="123456@someone.za">
  <Transaction mode="Request" transactionID="113">
    <GetUserProfileRequest userID="wv:doboz.bela@operator.za">
      <MetaInfo clientOriginated="Yes">
        <Requestor>
          <User userID="wv:quador@operator.hu">
            <ClientID url="http://www.wireless-village.org/applications/MudServer" />
          </User>
        </Requestor>
      </MetaInfo>
    </GetUserProfileRequest>
  </Transaction>
</Session>
</WV-SSP-Message>
```

```
<WV-SSP-Message xmlns="http://www.wireless-village.org/SSP1.0">
<Session sessionID="123456@operator.hu">
  <Transaction mode="Response" transactionID="113">
    <UserProfile>
      <Status code="200"/>
      <UserProfileValue userID="wv:doboz.bela@operator.za">
        <UPInfo attr="Services.Presence.Status">ON</UPInfo>
        <UPInfo attr="Services.Common.VHSE">pgsm.hu</UPInfo>
      </UserProfileValue>
    </UserProfile>
  </Transaction>
</Session>
</WV-SSP-Message>
```

UpdateUserProfile transaction

```
<WV-SSP-Message xmlns="http://www.wireless-village.org/SSP1.0">
<Session sessionID="123456@someone.za">
  <Transaction mode="Request" transactionID="113">
    <UpdateUserProfileRequest userID="wv:doboz.bela@operator.za">
      <MetaInfo clientOriginated="Yes">
        <Requestor>
          <User userID="wv:quador@operator.hu">
            <ClientID url="http://www.wireless-village.org/applications/MudServer" />
          </User>
        </Requestor>
      </MetaInfo>
      <UserProfileValue>
```

```
<UPInfo attr="Services.Presence.Status">ON</UPInfo>
<UPInfo attr="Services.Common.VHSE">pgsm.hu</UPInfo>
</UserProfileValue>
</GetUserProfileRequest>
</Transaction>
</Session>
</WV-SSP-Message>

<WV-SSP-Message xmlns="http://www.wireless-village.org/SSP1.0">
<Session sessionID="123456@operator.hu">
<Transaction mode="Response" transactionID="113">
<Status code="200"/>
</Transaction>
</Session>
</WV-SSP-Message>
```